REMARKS

A Summary of the Applicant-Initiated Examiner Interview that occurred on 11/08/2005 is provided hereinabove.

In the current and non-final Office Action dated 08/10/2005, claims 1-25, 37-45, and 49-56 were examined.

Claims 1-25, 37-45, and 49-56 were rejected.

Specifically:

Claims 1-25, 37-45, and 49-56 were "rejected under 35 U.S.C. 102(b) as being anticipated by Gong (WO 99/30217)."

Of pending claims 1-25, 37-45, and 49-56, six (6) claims 1, 24, 25, 37, 49, and 53 are independent. These six independent claims 1, 24, 25, 37, 49, and 53 are addressed below.

I. It is respectfully submitted that Gong does not anticipate (or render obvious) independent claim 1, 24, 25, or 37.

A. Gong does not describe or suggest the merging of permissions.

B. Thus, no art of record, either alone or in any combination, anticipates or renders obvious at least the following elements in conjunction with the other elements of their respective claims:

Claim 1: **generating the permission grant set by merging two or more code-group permission sets, each code-group permission set of the two or more code-group permission sets being associated with a code group in which the code assembly is a member.**

Claim 24: **generating the permission grant set by merging two or more code-group permission sets, each code-group permission set of the two or more code-group permission sets being associated with a code group in which the code assembly is a member.**

Claim 25: **generating the permission grant set by merging two or more code-group permission sets, each code-group permission set of the two or more code-group permission sets being associated with a code group in which the code assembly is a member.**

Claim 37: **generating the permission grant set by merging two or more code-group permission sets, each code-group permission set of the two or more code-group permission sets being**

associated with a code group in which the code assembly is a member.

II. It is respectfully submitted that Gong does not anticipate (or render obvious) independent claim 49 or 53.

A. On page 2 in the "Response to Arguments" section at paragraph no. 3, the current Office Action reads:

> Referring to the independent claims 49 and 53, Applicant argues that Gong does not address verification in any relationship with permission rights. With respect to this argument examiner points out that the instant claims do no recite that the verification is performed in "relationship with permission rights". Referring to the instant claims the limitation "performing verification on the code assembly" is met by Fig. 4 and the limitation "determining whether the code assembly may be executed despite the verification failure" is met by block 428 in Fig.4.

Gong indicates that its "Figure 4 is a flow chart showing the steps involved in implementing protection domains in accordance with an embodiment of the present invention;" It is respectfully submitted that this does not equate to **performing verification on the code assembly** (claim 49). If this rejection is maintained against claim 49, then an explanation is requested of how establishing a protection domain for a class and/or a mapping there between can possibly correspond to **performing verification on the code assembly** (claim 49).

Block 428 of Figure 4 of Gong reads, "Establish Mapping Of Class To Protection Domain?" It is noted that the question mark in block 428 appears to be a typographical error. Gong reads on page 12, at lines 21-24, "Next, in step 428, the mapping of the class to the protection domain is established. The mapping of the class to the protection domain is added to a mapping data structure maintained within the domain mapper 248. In this example, a mapping between class 260 and protection domain object 282 is created.". It is respectfully submitted that there is no

conditional determination; instead, a mapping is established. If this rejection is maintained against claim 49, then an explanation is requested as to how this can possibly correspond to **determining whether the code assembly may be executed despite the verification failure, responsive to the evaluating operation** (claim 49).

B. Gong only describes verification in the limited context of determining whether a class is to be considered an unsigned class.

By way of example, it appears that Gong addresses verification in only one paragraph:

> A class may be associated with the digital signature associated with the file used to transport code defining the class, or the class definition of the class may be specifically associated with a digital signature. A class that is associated with a valid digital signature is referred to as being signed. Valid digital signatures are digital signatures that can be verified by known keys stored in a database. If a class is associated with a digital signature which can not be verified, or the class is not associated with any digital signature, the class is referred to as being unsigned. Unsigned classes may be associated with a default key. A key may be associated with a name, which may be used to look up the key in the database.
>
> (WO 99/30217, Page 8, Lines 28-35.)

Thus, Gong does not describe **determining whether the code assembly may be executed despite the verification failure, responsive to the evaluating operation** (claim 49) or **determining based on the permission grant set that a step of a verification process is unnecessary** (claim 53).

C. Thus, no art of record, either alone or in any combination, anticipates or renders obvious at least the following elements in conjunction with the other elements of their respective claims:

Claim 49: **evaluating the evidence relative to the security policy . . . *performing verification on the code assembly* . . . detecting a verification failure of the code assembly in the operation of performing verification . . . *determining whether the code assembly may be executed despite the verification failure, responsive to the evaluating operation.***

Claim 53: **generating a permission grant set, responsive to the evaluating operation . . . *determining based on the permission grant set that a step of a verification process is unnecessary* . . . communicating to a verification module that the step of the verification process may be bypassed . . . performing the verification process on the code assembly with the verification module . . . *bypassing the step of the verification process, responsive to the communicating operation.***

III. It is respectfully submitted that Gong does not anticipate (or render obvious) dependent claims 2-23, 38-45, 50-52 and 54-56.

Reasons for the allowability of independent claims 1, 37, 49, and 53 have been provided above. Claims 2-23, 38-45, 50-52 and 54-56 depend from these independent claims 1, 37, 49, and 53, respectively.

Each dependent claim also includes additional element(s) militating toward allowability. For example, certain claims (e.g., claims 10-16 and 40-42) recite element(s) directed to a **policy level**. It is noted that Gong does not describe or suggest **policy level** as claimed.

Nevertheless, it is also respectfully submitted that the dependent claims are allowable at least for the reasons given above in connection with their respective independent claims.

CONCLUSION

It is respectfully submitted that all of the pending claims 1-25, 37-45, and 49-56 are allowable, and prompt action to that end is hereby requested.

Respectfully Submitted,

Date: 12/12/2005 By: Keith W. Saunders

Keith W. Saunders
Reg. No. 41,462
(509) 324-9256 x238